Metronome Documentation

Contents

[Feasibility Study 1](#_Toc123908378)

[Analysis 1](#_Toc123908379)

[Stakeholders 1](#_Toc123908380)

[Research 2](#_Toc123908381)

[Essential Features 2](#_Toc123908382)

[Limitations 2](#_Toc123908383)

[Design 2](#_Toc123908384)

[Implementation 2](#_Toc123908385)

[Testing 3](#_Toc123908386)

[Installation 3](#_Toc123908387)

[Installation 3](#_Toc123908388)

[Maintenance 3](#_Toc123908389)

# Feasibility Study

I would like to make a metronome. A metronome is a device which has a clock hand that bounces back and forth and the number of times this happens per minute can be changed. It is used by musicians and others to keep in time. We will make it web based and ensure it can be used on multiple devices.

I know the problem is solvable because calculating the number of sounds made per minute can be calculated in a finite number of steps and reasonable time. This is calculated by converting a BPM into a required delay using a theoretical approach.

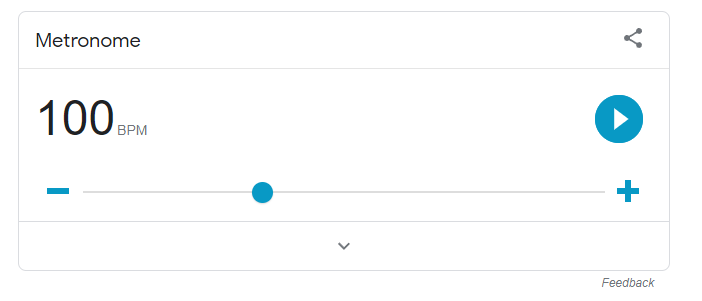
The budget is limited(£0) and we have 2 weeks to complete the project.

# Analysis

## Stakeholders

Henry is a 16-year-old who loves to play guitar on a weekly basis. He has a great interest in a metronome that could help to keep him in time when playing. Sometimes he may be using a mobile device but may also use a tablet or computer.

## Research

<https://www.google.com/search?q=google+metronome&rlz=1C1GCEA_enGB813GB815&oq=google+metronome&aqs=chrome..69i57j0i512l9.2804j0j1&sourceid=chrome&ie=UTF-8&safe=active&ssui=on>

Google provides a metronome which is free, easily acceptable and simple.

## Essential Features

My solution must allow you to set the BPM of the metronome, start/stop the beat and provide a visual implementation and audible noise every beat.

The solution must work on any web enabled device with a clear, touchscreen friendly interface

There should be limited text so it’s suitable for anyone.

It should be freely accessible online.

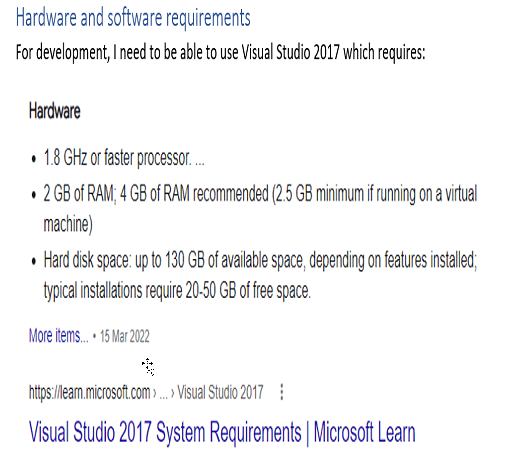
## Limitations

There is insufficient time to add additional features but these could be added later. My solution will not allow you tom compare your playing to the beat, there will be no memory of users and no tracked data. A feature that could be later added would also be a customisable BPM over a set time period allowing musicians to practice specific pieces or techniques.

There is no requirement to make a native android or apple app: the metronome will be purely web based in order to make it freely available to as many people as possible.

## Hardware and Software Requirements

For development, I need to be able to use Visual Studio 2017 which requires:



<https://learn.microsoft.com/en-us/visualstudio/releases/2017/vs2017-system-requirements-vs>

For the sake of the stakeholders to run the metronome they need it to run on the latest version of chrome available on tablets, phones and PCs.

My stakeholder has an android mobile phone

|  |  |
| --- | --- |
| Screen size: |  |
| OS: |  |
| Storage: |  |
| RAM: |  |

## Success Criteria

|  |  |  |
| --- | --- | --- |
| Number | Criteria | Justification |
| 1 | Must be accessible on an android 12 device on chrome | Stakeholder has an android 12 device with a reliable WiFi connections. Any updates can be automatically rolled out |
| 2 | User can set the BPM | The stakeholder needs to modify BPM to practice between 50-120BPM |
| 3 | Any invalid BPM will be automatically corrected to the closest sensible value between 50-120BPM | The stakeholder wants to focus on playing guitar and might accidentally type in invalid data |
| 4 | There should be a start and stop button that toggles when you press it | The user should be able to control the metronome with their finger with a really simple UI |
| 5 | The user should be able to increase the BPM by 5 using a touch button | The stakeholder needs to be able to easily modify the speeds by a common increase so they can practice different speeds |
| 6 | The user should be able to decrease the BPM by 5 using a touch button |
| 7 | There should be an audible beep every beat | If they are focusing on sheet music or the instrument they will want an audible cue to play alongside |
| 8 | There should be a visual cue every beat | People may not have hearing difficulties or not want to be using sound so require a visual cue |
| 9 | There should be a high contrast mode button which toggles a black and white colour scheme with larger text | Sometimes the metronome will be on a small screen or far away and users will need an easier visual experience so they don’t mistake the numbers |

# Design

## Algorithms

The user should be able to enter the BPM score. The browser needs to know how long to pause between each tick in milliseconds. This can be done using the following algorithm

BPM = user input  
interval = 60000 / user input

## Usability features

**-**

**+**

BPM

60

## Variables and validation

|  |  |  |
| --- | --- | --- |
| Variable | Data type | Validation |
| BPM | Integer | Range check(between 50-120) |
| BPMUserInput | String | Presence check and Type check |
| IntervalMS | Integer | Make sure it’s rounded to 0 dp |
| HighContrastMode | Boolean | Checkbox can be ticked or not ticked |
| IsPlaying | Boolean | Either true or false |

# Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Number** | **Description** | **Success Criteria** | **Test Data** | **Expected Result** |
| **1** | **Basic web page** | **1** | **Page load** | **A web page with a title and text box to enter the BPM score displays in chrome** |
| **2** | **BPM** | **2** | **120** | **Valid data accepted** |
| **3** | **BPM** | **2** | **50** | **Valid data accepted** |
| **4** | **BPM** | **2** | **60** | **Valid data accepted** |
| **5** | **BPM** | **3** | **130** | **BPM set to 120** |
| **6** | **BPM** | **3** | **30** | **BPM set to 50** |
| **7** | **BPM** | **3** | **“”** | **BPM set to 50** |
| **8** | **BPM** | **3** | **Potato** | **BPM set to 50** |
| **9** | **Increasing BPM** | **5** | **BPM set to 60 Press increase** | **BPM increases up to 65** |
| **10** | **Increasing BPM** | **5** | **BPM set to 120 Press increase** | **BPM stays at 120** |
| **11** | **Increase BPM** | **5** | **BPM set to 119** **Press increase** | **BPM increases to 120** |
| **12** | **Decreasing BPM** | **6** | **BPM set to 60 Press decrease** | **BPM stays at to 60** |
| **13** | **Decreasing BPM** | **6** | **BPM set to 120 Press decrease** | **BPM decreases to 115** |
| **14** | **Decreasing BPM** | **6** | **BPM set to 61** **Press decrease** | **BPM decreases to 60** |
| **15** | **Playing noise** | **7** | **BPM set to 60** | **60 beats should play in a minute at regular intervals** |
| **16** | **Playing noise** | **7** | **BPM set to 120** | **120 beats should play in a minute at regular intervals** |
| **17** | **Playing noise** | **7** | **BPM set to 77** | **77 beats should play in a minute at regular intervals** |
| **18** | **Button to start** | **4** | **Button is paused** **Button is clicked** | **Should change to stop** |
| **19** | **Button to stop** | **4** | **Button is started** **Button is clicked** | **Should change to start** |

## Post development Testing

Here are some questions to give my stakeholders after I’ve finished developing the solution:

Functionality  
What did you want the program to do? What did you click on (exactly, in what order?) What happened? Did the program do everything you wanted it to do? Were there any features that you wished were there but weren’t?

Robustness  
Did the program crash? If so, when (what did you do immediately before? What were you trying to do?)

Usability

# Implementation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.

# Installation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.

# Installation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.

# Maintenance

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.